# HUSKY TECHNICAL NOTE

Number 265

Issue 2 Date 04MAY89

Applicable to: HUSKY HUNTER, SP AND M-208.

Abstract: DESCRIPTION OF THE INTERFACE REQUIRED TO UTILISE THE ORACLE GT IN APPLICATIONS WITH THE HUSKY HUNTER.

Title: USE OF ORACLE GT WITH APPLICATION PROGRAMS FOR THE HUSKY HUNTER.

## 1. INTRODUCTION

This technical note describes the interface requirements to utilise the Oracle GT with Husky BASIC application programs. The disk functions will execute from the BASIC transient program area and therefore require space within this area to execute from.

Inorder to provide this space the BASIC MAXFILES facility is used to reserve an area of the transient program area for storing the interface software.

It is important that this technical note be studied carefully inorder to gain a thorough understanding prior to attempting to utilise the interface in an application. The source files and demonstration files described are supplied on the Oracle GT system disk.

### 1.1 Facilities offered by the Oracle GT

The Oracle GT provides an enhanced form of storage media to the prior Oracle disk drive. It provides a means of storing and retrieving files from a disk which is compatible with the standard IBM 3.5 inch format. This provides two disk formats, either 720K byte capacity or 1.4M byte capacity. This facility provides a convenient means of exchanging files between the Husky Hunter and many desktop personal computers, as well as providing a very large capacity back-up media to the Husky Hunter.

Important

The disk format used by the Oracle Gt is not compatible with that of the previous Oracle. Disks are therefore not transferable between the two drives.

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system, in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 682383. Telex: 317450 Husky G.

It is important also that the correct quality disks are chosen, particularly for the 1.4M capacity format. Use of inferior disks is a false economy which will lead to data loss and file corruption.

## 2. ORACLE GT DISK FUNCTIONS

The interface software provides six functions for controlling the Oracle GT:

a) Disk directory
b) File load
c) File dump
d) File erase
e) Rename file
f) Disk format

The functions (a), (b) and (c) are capable of supporting wild cards, so that groups of files may be copied from or written to the disk drive if required.

### 2.1 The MAXFILES Function

The BASIC function MAXFILES is used to specify in a BASIC program the number of files that can be opened simultaneously during an application. Each file which is opened reserves 168 bytes from the transient program area (TPA).

The function MAXFILES can be used to reserve areas of memory for use by machine code.

This technique is described in technical note TN175. The Oracle GT interface software utilises this facility.

Inorder to use this interface software the following statement must be executed in the application.

```
MAXFILE = 15
```

This means that an application can only have two files opened simultaneously, and these files must use filenumbers £1 and £2 respectively. Filenumbers greater than £2 must not be used, as this will destroy the disk software. The machine code is loaded starting at address CE77H (52855D).

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder. Husky Computers Limited

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

## 2.2 Entering the Interface Software

The interface machine code can be entered by a series of BASIC POKE statements. These statements need only be executed once in an application, as the machine code is stored above the normal program area. Alternatively a loader program could be used to copy the machine code from a file in the ram disk area of the Hunter into the storage area if this is more convenient.

**POKE.BAS**

```
9000 REM The POKE statements to load the machine code
10000 POKE52855,34,92,214,121,67,61,135,111
10010 POKE52863,38,0,17,145,206,25,126,35
10020 POKE52871,102,111,229,197,205,66,213,193
10030 POKE52879,225,233,157,206,144,209,112,210
10040 POKE52887,115,211,158,211,238,211,205,246
10050 POKE52895,212,205,26,213,175,50,181,214
10060 POKE52903,205,71,207,218,115,213,167,194
10070 POKE52911,215,206,17,28,209,205,68,212
10080 POKE52919,205,52,209,218,115,213,34,88
10090 POKE52927,214,205,21,208,175,50,181,214
10100 POKE52935,33,169,214,205,106,207,218,115
10110 POKE52943,213,167,194,215,206,195,183,206
10120 POKE52951,30,82,205,36,212,30,47,205
10130 POKE52959,36,212,205,143,207,218,115,213
10140 POKE52967,30,79,32,2,30,87,205,36
10150 POKE52975,212,30,32,205,36,212,205,135
10160 POKE52983,212,30,70,205,24,212,205,140
10170 POKE52991,212,205,90,212,56,20,167,32
10180 POKE52999,17,205,141,212,205,146,212,56
10190 POKE53007,9,111,205,146,212,56,3,103
10200 POKE53015,24,3,33,0,0,203,37,203
10210 POKE53023,20,17,94,214,205,217,207,33
10220 POKE53031,54,207,1,17,0,237,176,62
10230 POKE53039,36,18,205,177,208,175,201,75
10240 POKE53047,32,98,121,116,101,115,32,97
10250 POKE53055,118,97,105,108,97,98,108,101
10260 POKE53063,205,135,212,30,83,205,76,212
10270 POKE53071,216,167,192,205,141,212,33,128
10280 POKE53079,224,6,32,205,146,212,216,119
10290 POKE53087,35,16,248,42,144,224,34,74
10300 POKE53095,214,175,201,229,205,135,212,30
10310 POKE53103,78,205,24,212,225,35,6,12
10320 POKE53111,205,123,212,237,91,74,214,213
10330 POKE53119,90,205,24,212,209,205,24,212
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
10340 POKE53127,205,140,212,205,90,212,24,192
10350 POKE53135,205,135,212,30,81,205,24,212
10360 POKE53143,205,140,212,205,146,212,245,205
10370 POKE53151,146,212,48,3,241,55,201,241
10380 POKE53159,230,32,201,6,8,205,181,207
10390 POKE53167,62,46,18,19,6,3,126,230
10400 POKE53175,127,18,19,35,16,248,201,62
10410 POKE53183,63,6,11,190,200,35,16,251
10420 POKE53191,62,1,167,201,213,235,33,0
10430 POKE53199,0,167,40,4,25,61,32,252
10440 POKE53207,209,201,175,1,240,216,205,254
10450 POKE53215,207,1,24,252,205,254,207,1
10460 POKE53223,156,255,205,254,207,1,246,255
10470 POKE53231,205,254,207,1,255,255,205,254
10480 POKE53239,207,192,62,48,18,19,201,213
10490 POKE53247,22,255,229,221,225,20,9,56
10500 POKE53255,249,221,229,225,66,209,176,200
10510 POKE53263,62,48,128,18,19,201,33,94
10520 POKE53271,214,6,40,54,32,35,16,251
10530 POKE53279,42,88,214,229,235,106,38,0
10540 POKE53287,62,128,205,203,207,22,0,25
10550 POKE53295,229,17,94,214,62,255,213,205
10560 POKE53303,218,207,235,227,205,184,208,209
10570 POKE53311,62,32,18,19,235,227,205,10
10580 POKE53319,209,40,1,35,203,69,40,1
10590 POKE53327,35,209,213,62,255,205,218,207
10600 POKE53335,235,227,205,184,208,209,62,75
10610 POKE53343,18,19,225,213,108,38,0,35
10620 POKE53351,62,255,205,218,207,235,227,205
10630 POKE53359,184,208,225,54,32,35,54,82
10640 POKE53367,35,54,47,35,54,87,229,33
10650 POKE53375,137,224,203,126,225,40,2,54
10660 POKE53383,79,35,54,32,35,54,58,35
10670 POKE53391,235,33,129,224,229,221,225,221
10680 POKE53399,203,9,126,40,4,62,40,18
10690 POKE53407,19,205,170,207,221,203,9,126
10700 POKE53415,40,3,62,41,18,62,36,50
10710 POKE53423,134,214,17,94,214,175,195,150
10720 POKE53431,213,6,4,126,254,48,192,54
10730 POKE53439,32,35,16,247,201,229,213,197
10740 POKE53447,205,191,212,167,194,174,212,14
10750 POKE53455,11,205,5,0,167,40,241,14
10760 POKE53463,1,205,5,0,254,3,32,232
10770 POKE53471,193,209,225,62,255,55,201,205
10780 POKE53479,196,208,218,170,212,205,182,212
10790 POKE53487,15,15,15,15,71,205,196,208
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.



Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
10800 POKE53495,218,170,212,205,182,212,128,71
10810 POKE53503,205,196,208,254,13,194,170,212
10820 POKE53511,120,167,201,125,230,7,245,203
10830 POKE53519,60,203,29,203,60,203,29,203
10840 POKE53527,60,203,29,241,201,32,82,101
10850 POKE53535,99,115,32,32,66,121,116,101
10860 POKE53543,115,32,32,69,120,116,32,65
10870 POKE53551,99,99,13,10,36,42,74,214
10880 POKE53559,34,90,214,33,128,224,17,134
10890 POKE53567,214,1,32,0,237,176,221,33
10900 POKE53575,134,214,221,126,12,50,86,214
10910 POKE53583,221,126,15,50,87,214,62,63
10920 POKE53591,50,146,214,33,134,214,205,106
10930 POKE53599,207,216,167,32,26,221,33,128
10940 POKE53607,224,58,86,214,221,190,12,48
10950 POKE53615,234,221,126,12,50,86,214,221
10960 POKE53623,126,15,50,87,214,24,215,58
10970 POKE53631,86,214,103,58,87,214,111,229
10980 POKE53639,42,90,214,34,74,214,225,167
10990 POKE53647,201,205,246,212,205,26,213,175
11000 POKE53655,50,181,214,205,71,207,218,115
11010 POKE53663,213,167,194,105,213,205,52,209
11020 POKE53671,218,115,213,235,106,38,0,62
11030 POKE53679,128,205,203,207,22,0,25,34
11040 POKE53687,88,214,205,28,212,167,40,10
11050 POKE53695,205,32,212,254,3,32,243,195
11060 POKE53703,130,213,33,128,224,17,134,214
11070 POKE53711,1,13,0,237,176,175,50,146
11080 POKE53719,214,17,134,214,205,48,212,17
11090 POKE53727,134,214,205,52,212,60,202,125
11100 POKE53735,213,205,38,213,33,0,0,34
11110 POKE53743,83,214,205,135,212,30,82,205
11120 POKE53751,24,212,58,74,214,95,205,24
11130 POKE53759,212,237,91,83,214,213,205,24
11140 POKE53767,212,209,90,205,24,212,205,140
11150 POKE53775,212,205,90,212,218,115,213,167
11160 POKE53783,194,115,213,205,141,212,33,128
11170 POKE53791,224,14,0,6,128,205,146,212
11180 POKE53799,218,115,213,119,129,79,35,16
11190 POKE53807,244,205,146,212,218,115,213,129
11200 POKE53815,60,194,115,213,17,134,214,205
11210 POKE53823,56,212,167,194,135,213,42,83
11220 POKE53831,214,35,237,91,88,214,167,237
11230 POKE53839,82,25,194,238,209,17,134,214
11240 POKE53847,205,60,212,205,161,213,175,50
11250 POKE53855,181,214,33,169,214,205,106,207
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
11260 POKE53863,218,115,213,167,202,164,209,175
11270 POKE53871,201,205,246,212,17,134,214,205
11280 POKE53879,8,212,60,202,105,213,42,80
11290 POKE53887,248,34,74,214,42,82,248,34
11300 POKE53895,76,214,205,28,212,167,40,10
11310 POKE53903,205,32,212,254,3,32,243,195
11320 POKE53911,130,213,33,128,224,205,29,213
11330 POKE53919,205,43,213,205,143,207,218,115
11340 POKE53927,213,194,145,213,175,50,181,214
11350 POKE53935,50,201,214,17,169,214,205,16
11360 POKE53943,212,60,202,110,213,205,135,212
11370 POKE53951,30,68,205,76,212,218,115,213
11380 POKE53959,254,167,202,120,213,205,135,212
11390 POKE53967,30,77,205,76,212,218,115,213
11400 POKE53975,167,194,125,213,205,141,212,205
11410 POKE53983,146,212,218,115,213,111,38,0
11420 POKE53991,34,78,214,33,0,0,34,83
11430 POKE53999,214,17,169,214,205,20,212,167
11440 POKE54007,194,67,211,205,135,212,30,87
11450 POKE54015,205,24,212,58,78,214,95,205
11460 POKE54023,24,212,237,91,83,214,213,205
11470 POKE54031,24,212,209,90,205,24,212,205
11480 POKE54039,140,212,205,90,212,218,115,213
11490 POKE54047,167,194,125,213,33,128,224,6
11500 POKE54055,128,175,94,131,245,197,229,205
11510 POKE54063,24,212,225,193,241,35,16,242
11520 POKE54071,47,95,205,24,212,42,83,214
11530 POKE54079,35,195,237,210,205,135,212,30
11540 POKE54087,67,205,24,212,205,140,212,205
11550 POKE54095,90,212,205,161,213,42,74,214
11560 POKE54103,34,80,248,42,76,214,34,82
11570 POKE54111,248,33,134,214,17,92,224,1
11580 POKE54119,13,0,237,176,205,12,212,60
11590 POKE54127,200,195,125,210,205,246,212,33
11600 POKE54135,135,214,205,190,207,202,140,213
11610 POKE54143,205,26,213,205,143,207,218,115
11620 POKE54151,213,194,145,213,205,135,212,30
11630 POKE54159,68,205,76,212,218,115,213,254
11640 POKE54167,167,202,120,213,195,52,207,205
11650 POKE54175,246,212,42,92,214,126,35,102
11660 POKE54183,111,17,12,0,25,17,151,214
11670 POKE54191,1,11,0,237,176,33,135,214
11680 POKE54199,205,190,207,202,140,213,33,151
11690 POKE54207,214,205,190,207,202,140,213,205
11700 POKE54215,143,207,218,115,213,194,145,213
11710 POKE54223,205,135,212,30,76,205,24,212
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
11720 POKE54231,33,135,214,6,31,94,229,197
11730 POKE54239,205,24,212,193,225,35,16,245
11740 POKE54247,205,140,212,205,90,212,201,197
11750 POKE54255,205,135,212,30,88,205,24,212
11760 POKE54263,241,95,205,24,212,205,140,212
11770 POKE54271,205,230,208,56,2,167,200,55
11780 POKE54279,201,14,17,24,58,14,18,24
11790 POKE54287,54,14,15,24,50,14,20,24
11800 POKE54295,46,14,4,24,42,14,11,24
11810 POKE54303,38,14,47,24,34,14,2,24
11820 POKE54311,30,14,3,24,26,14,42,24
11830 POKE54319,22,14,19,24,18,14,22,24
11840 POKE54327,14,14,21,24,10,14,16,24
11850 POKE54335,6,14,38,24,2,14,9,229
11860 POKE54343,205,5,0,225,201,205,24,212
11870 POKE54351,6,12,33,170,214,205,123,212
11880 POKE54359,205,140,212,205,146,212,56,75
11890 POKE54367,205,182,212,15,15,15,15,71
11900 POKE54375,205,146,212,56,62,205,182,212
11910 POKE54383,128,71,205,146,212,254,13,32
11920 POKE54391,50,120,167,201,94,229,197,205
11930 POKE54399,24,212,193,225,35,16,245,201
11940 POKE54407,30,5,195,24,212,0,30,13
11950 POKE54415,195,24,212,229,213,197,33,0
11960 POKE54423,64,6,1,205,191,212,167,32
11970 POKE54431,14,43,125,180,32,245,16,243
11980 POKE54439,193,209,225,62,255,55,201,205
11990 POKE54447,199,212,193,209,225,167,201,254
12000 POKE54455,58,56,2,214,55,230,15,201
12010 POKE54463,229,197,213,205,44,212,24,6
12020 POKE54471,229,197,213,205,40,212,209,193
12030 POKE54479,225,201,17,128,0,58,104,0
12040 POKE54487,71,33,0,0,167,40,3,25
12050 POKE54495,16,253,58,201,214,95,22,0
12060 POKE54503,25,201,197,126,230,127,95,205
12070 POKE54511,36,212,193,35,16,244,201,33
12080 POKE54519,134,214,17,135,214,1,35,0
12090 POKE54527,54,0,237,176,42,92,214,126
12100 POKE54535,35,102,111,17,135,214,1,12
12110 POKE54543,0,237,176,175,50,134,214,50
12120 POKE54551,146,214,201,33,134,214,17,169
12130 POKE54559,214,1,13,0,237,176,201,33
12140 POKE54567,135,214,24,3,33,170,214,6
12150 POKE54575,8,205,233,212,30,46,205,36
12160 POKE54583,212,6,3,205,233,212,30,32
12170 POKE54591,195,36,212,33,0,0,34,93
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damages resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
12180 POKE54599,248,17,248,247,33,91,213,1
12190 POKE54607,14,0,237,176,62,128,50,75
12200 POKE54615,248,195,64,212,9,9,1,1
12210 POKE54623,1,0,0,3,3,0,0,0
12220 POKE54631,0,0,17,175,213,24,38,17
12230 POKE54639,186,213,24,33,17,206,213,24
12240 POKE54647,28,17,220,213,24,23,17,238
12250 POKE54655,213,24,18,17,2,214,24,13
12260 POKE54663,17,13,214,24,8,17,37,214
12270 POKE54671,24,3,17,55,214,26,19,245
12280 POKE54679,14,9,205,5,0,205,161,213
12290 POKE54687,241,201,30,13,14,2,205,5
12300 POKE54695,0,30,10,14,2,195,5,0
12310 POKE54703,177,45,32,110,111,32,102,105
12320 POKE54711,108,101,36,178,45,32,99,97
12330 POKE54719,110,110,111,116,32,111,112,101
12340 POKE54727,110,32,102,105,108,101,36,179
12350 POKE54735,45,32,100,105,115,107,32,101
12360 POKE54743,114,114,111,114,36,180,45,32
12370 POKE54751,102,105,108,101,32,114,101,97
12380 POKE54759,100,32,111,110,108,121,36,181
12390 POKE54767,45,32,99,97,110,110,111,116
12400 POKE54775,32,109,97,107,101,32,102,105
12410 POKE54783,108,101,36,182,45,32,97,98
12420 POKE54791,111,114,116,101,100,36,183,45
12430 POKE54799,32,99,97,110,110,111,116,32
12440 POKE54807,119,114,105,116,101,32,116,111
12450 POKE54815,32,102,105,108,101,36,184,45
12460 POKE54823,32,102,105,108,101,110,97,109
12470 POKE54831,101,32,101,114,114,111,114,36
12480 POKE54839,185,45,32,100,105,115,107,32
12490 POKE54847,114,101,97,100,32,111,110,108
12500 POKE54855,121,36,0,84,133,112,0,133
12510 RETURN
```

## 2.3 Passing Parameters to/from the Interface

Parameters must be passed between the Hunter BASIC and the disk utility software. The following parameters must be passed:-

i) the required disk function number
ii) the filename(s) where applicable
iii) the status code variable
iv) a selection code

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G.

## 2.4 Passing the Status Code Variable

When a disk operation has been carried out and an error has occurred, an error code is returned to the calling program.

Passing of the status code variable to the Husky BASIC application program is achieved by using the CALL function:-

```
ER = CALL(N)
```

where ER is the status code variable

N is the entry address to interface software

## 2.5 Passing the Disk Function Number

The disk function number is simply a number assigned to a disk function for the purpose of identification.

The assigned function numbers are as follows:-

| Disk Function | Function Number |
|---|---|
| **Disk Directory** | **1** |
| **File Load** | **2** |
| **File Dump** | **3** |
| **Erase File** | **4** |
| **Rename File** | **5** |
| **Disk Format** | **6** |

The disk function number is passed to the disk software by use of the Hunter BASIC ARG function:-

```
A = ARG(N)
```

where N is the required disk function number

## 2.6 Passing Filenames

Filenames are passed to the disk software via the Hunters stack.

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited

Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

This is achieved by pushing the pointer of the string variable containing the filename onto the stack prior to calling the disk routine.

```
F$ = "FILENAMEEXT"
F  = VARPTR (F$)
PUSH F
```

It is important to note that the filename must be in the correct CP/M format. The disk software will interpret this filename literally.

The name must contain valid ascii characters acceptable for a filename, and must be in upper case alpha characters.

On return from the disk software to the Hunter BASIC application the stack must be corrected by 'popping' the address off the stack.

```
F = pop(1)
```

If this statement is not performed the BASIC variable stack will accumulate addresses and eventually overflow reporting a fatal stack error.

## 2.7 Passing the Selection Code

The disk FORMAT function requires a parameter to select the capacity of disk to format. This parameter is passed to the application via the BASIC ARG parameter.

For example to format a 720K capacity disk requires the selection code to be set to '1'.

This parameter is transferred by using the argument:

```
A = ARG (6 + 256 * 1)
```

where the '1' selects 720K capacity
and the '6' the format function

Inorder to format a 1.4M capacity disk the selection parameter should be '0'

```
A = ARG (6 + 256 * 0)
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying, or otherwise, without prior permission from the copyright holder Husky Computers Limited.

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damages resulting from the use of the information contained within this document.



Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

## 3. EXECUTING DISK FUNCTIONS

The following section will provide demonstration models of how each of the disk functions may be utilised by a BASIC application program.

### 3.1 Obtaining a Disk Directory

The following program demonstrates how a directory may be obtained of an Oracle GT disk. On execution of the program the names of all files on the disk will be displayed on the Hunter screen.

DIRECT.BAS

```
1 REM Demonstration of the DIRECTORY function
5 maxfiles=15:gosub 10000
10 f$="???????????"
20 a=ARG(1):REM select directory of Oracle GT
30 f=VARPTR(f$)
40 PUSHf
45 CLS
50 PRINT"=============* DIRECTORY *================"
60 er=CALL(52855)
70 PRINT"=============* completed *================"
80 a=POP(1)
90 if er <> 0 then beep: print"Error Code: ",er:end
100 print "Operation completed without error":end
```

Note that the string variable f$ contains the character '?' in all locations of the name. This means that any filename will match with this name. A specific file or group of files may also be specified if required.

For example to obtain a directory of all .COM files on a disk the string variable f$ should be equated as follows:

```
f$ = "        COM"
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

## 3.2 Loading Files from the Oracle GT

The following program demonstrates how all .COM files may be copied from the Oracle GT into the Hunter memory.

**LOAD.BAS**

```
1 REM Demonstration of the LOAD function
5 maxfiles=15:gosub 10000
10 f$="????????COM"
20 a=ARG(2):REM select load from Oracle GT
30 f=VARPTR(f$)
40 PUSHf
45 CLS
50 PRINT"==============*  LOADING  *================"
60 er=CALL(52855)
70 PRINT"==============* completed *================"
80 a=POP(1)
90 if er <> 0 then beep:print "Error Code: ",er:end
100 print "Operation completed without error":end
```

Note that this function supports the use of wildcard characters '?' in the string variable f$ to specify groups of files to be copied.

## 3.3 Copying Files to the Oracle GT

This demonstration program shows how a group of files with the name DISK can be copied from the Hunter ram disk onto the Oracle GT.

**DUMP.BAS**

```
1 REM Demonstration of the DUMP function
5 maxfiles=15:gosub 10000
10 f$="DISK    ???"
20 a=ARG(3):REM select dump to Oracle GT
30 f=VARPTR(f$)
40 PUSHf
45 CLS
50 PRINT"==============*  COPYING  *================"
60 er=CALL(52855)
70 PRINT"==============* completed *================"
80 a=POP(1)
90 if er <> 0 then print "Error Code: ",er:end
100 print "Operation completed without error":end
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited

Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

## 3.4 Erasing Files from the Oracle GT

Files can be erased from the Oracle **GT** using the erase file function.

The erase functions does **not** support the use of wild cards. The filename specified in the string variable **must** be specified.

This example shows how the file DISCO.ASM can be deleted from the Oracle GT disk.

**ERASE.BAS**

```
1 REM Demonstration of the ERASE function
5 maxfiles=15:gosub 10000
10 f$="DISCO   ASM"
20 a=ARG(4):REM select erase file from Oracle GT
30 f=VARPTR(f$)
40 PUSHf
45 CLS
50 PRINT"====* ERASING: "+f$ + " *===="
60 er=CALL(52855)
80 a=POP(1)
90 if er <> 0 then beep:print "Error Code: ",er:end
100 print "Operation completed without error":end
```

Inorder to minimise the machine code this function will not indicate an error if the file to be erased does not exist.

## 3.5 Renaming a File on the Oracle GT

This demonstration program shows how to rename a file OLDFILE.EXT on the Oracle GT to the name NEWFILE.EXT.

**RENAME.BAS**

```
1 REM Demonstration of the RENAME function
5 maxfiles=15:gosub 10000
6 DIM f$(0,40)
10 f$="OLDFILE EXT"+"!"+"NEWFILE EXT"
20 a=ARG(5):REM select rename function
30 f=VARPTR(f$)
40 PUSHf
45 CLS
50 PRINT"==========*  RENAMING  *================"
60 er=CALL(52855)
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited



Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
70 PRINT"==============*  completed *================"
80 a=POP(1)
90 if er <> 0 then print "Error Code: ",er:end
100 print "Operation completed without error":end
```

Note how the two filenames are specified in the string variable f$, the two names separated by the character '!'. Wildcards are **not** supported by this function.

### 3.6 Formatting a Disk on the Oracle GT

This program demonstrates how to format an Oracle GT disk using the disk FORMAT function.

The example below shows how the ARG statement is used to format a disk to 720K capacity.

**FORMAT.BAS**

```
1 REM Demonstration of the FORMAT function
10 maxfiles=15:gosub 10000
20 a=ARG(6+256*1):REM select format of 720K capacity disk
45 CLS
50 PRINT"==============* FORMATTING *================"
60 er=CALL(52855)
70 PRINT"==============* completed  *================"
90 if er <> 0 then beep:print "Error Code: ",er:end
100 print "Operation completed without error":end
```

The Oracle GT can support two capacities of disk. By altering the ARG statement as shown below 1.4M capacity disks may also be formatted.

```
20 a=ARG (6+256*0):REM select format, of 1.4M capacity disk
```

## 4. ERROR HANDLING

The status code variable is used by the disk software to report any error conditions that may arise during a disk operation to a controlling program.

The error message is returned onto the screen, and any progress information such as filenames being transferred parameters etc.

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise without prior permission from the copyright holder Husky Computers Limited

Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

The following error codes and their associated messages are shown below.

**00 - No error detected operation completed.**

**B1 - no file**

Search has failed to detect the file required.

**B2 - cannot open file**

Access to open the source file has been refused. This is usually due to an error in the file itself.

**B3 - disk error**

A timeout has occurred in the disk access. Check the connections are correct, a disk is in the drive or the baud rate is set to 4800 baud.

**B4 - file read only**

An attempt has been made to delete or write to a file with its read only attribute set.

**B5 - cannot make file**

A failure has occurred in creating a destination file. This could be due to either insufficient directory space or the destination drive is set to read only.

**B6 - aborted**

Use of control-C to abort the function call will usually result in this message.

**B7 - cannot write to file**

A failure has occurred in writing to the file probably due to insufficient space on the disk or the disk is set to read-only.

**B8 - filename error**

Wildcard characters '?' have be used in the filename which is not allowed in the disk function.

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise without prior permission from the copyright holder, Husky Computers Limited

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damages resulting from the use of the information contained within this document



Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone Coventry (0203) 668181 Telex 317450 Husky G

B9 - disk read only

The write protect toggle on the disk is set to read-only and an attempt has been made to write to the disk.

In addition to these software generated errors, Oracle GT disk errors may be returned in some instances. These additional errors which are generated by the FDC are listed below:-

01H - Incorrect drive specifier
02H - Missing address marker
03H - Disk is write protected
04H - Record not found
08H - DMA over-run error
09H - Attempted DMA over 64K segment
10H - CRC error
20H - Floppy disk controller error
30H - Disk drive door status change
40H - Seek error
80H - Drive not ready

The following errors are generated by the controlling software in the disk drive itself:-

A0H - Failed to format track
A1H - Directory space full
A2H - FAT space is full
A3H - File not found or failed to open for access
A4H - Read error
A5H - Write error
A6H - Close error
A7H - Read only access file

## 5. USE WITH MACHINE OR COMPILED APPLICATIONS

Supplied with the system file for the Oracle GT disk drive is the actual source file for the interface software this technical note describes.

In addition the next section of this note contains an assembly source listing of the same program.

For assembly applications there is no necessity for the code to be located in common memory, if required this could be relocated to a more convenient location in RAM.

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited



Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

## 5.1 Language Information

The disk software is written in HD64180 assembly language. The development package used to produce the code was AVOCET AVMAC 180 CROSS-ASSEMBLER VER 2.00 running on an IBM AT.

## 5.2 Utilising the disk functions

Depending on the application, not all of the disk functions may be required. It can be seen that the source file can be easily split to create a file with only those functions which are required. This file may then be linked into the main application source code or compiled code.

## 6. SOURCE LISTING OF DISK INTERFACE CODE

The following is an assembly source listing of the disk interface software written in HD64180 assembly code.

```
;*****************************************************************
;*                                                               *
;*                      HUNDISK FILE                             *
;*                                                               *
;*                  Ver: 2 December 1988                         *
;*                     John Battensby                            *
;*                                                               *
;*****************************************************************
;
;
;
;01.11.88 jjb file created
;04.12.88 jjb update to fix various faults
;27.04.89 jjb changes to perform stat of file before load function
;03.05.89 jjb further changes to correct load function
;
;
cr:       EQU     0DH             ;carriage return
lf:       EQU     0AH             ;line feed
brk:      EQU     27              ;break code
;
; specific equates used
;
dma:      equ     0E080h
fcbdr:    equ     005Ch           ;default fcb buffer
fcbfn:    equ     005DH
fcbex:    equ     0068H
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited

Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
fcbt1:    equ     0065H
fcbd0:    equ     006CH
system:   equ     5h

lstdir:   equ     0F850H
lstpge:   equ     0F852H


          DEFSEG HDOS,ABSOLUTE
          SEG   HDOS

          ORG     0CE77H          ;maxfiles=15 with 2 user files

start:    ld      (stack),hl      ;save the stack pointer address
          ld      a,c
          ld      b,e             ;preserve e register
          dec     a
          add     a,a
          ld      l,a
          ld      h,0
          ld      de,jump_tab
          add     hl,de           ;form address to function
          ld      a,(hl)
          inc     hl
          ld      h,(hl)
          ld      l,a
          push    hl
          push    bc
          call    setcom          ;ensure comms port set up
          pop     bc
          pop     hl
          jp      (hl)            ;go execute the function

jump_tab: dw      kerdir          ;directory function
          dw      kerload         ;file load function
          dw      kerdump         ;file dump function
          dw      kerera          ;file erase function
          dw      kerren          ;file rename function
          dw      kerform         ;disk format function
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise without prior permission from the copyright holder. Husky Computers Limited



Disclaimer. Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181. Telex: 317450 Husky G

```
$eject
;====================================
; DISK DIRECTORY FUNCTION
;====================================


kerdir:   call    initfcb
          call    move            ;transfer name to tbuf1

; search for the file
          xor     a
          ld      (tbuf1+12),a
          call    search1         ;look for first extent
          jp      c,dferr3        ;if timeout branch
          and     a
          jp      nz,dskfree      ;if no further files branch
; output the header
          ld      de,header
          call    prnstrg
; perform the calulation of number of records in the file
again:    call    filesize
          jp      c,dferr3        ;if error then branch
          ld      (workr11),hl
          call    prnlne          ;output file size
; restore first extent position
          xor     a
          ld      (tbuf1+12),a    ;clear extent
          ld      hl,tbuf1
          call    search2
          jp      c,dferr3        ;if timeout branch
          and     a
          jp      nz,dskfree      ;if no more give disk capacity
          jp      again           ;loop for next file

; output the disk space available

dskfree:  ld      e,'R'
          call    serout
          ld      e,'/'
          call    serout
; determine if disk is read only or not
          call    tstdskro
          jp      c,dferr3        ;if timeout branch
          ld      e,'O'
          jr      nz,dskf1
          ld      e,'W'
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited PO Box 35 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
dskf1:    call    serout
          ld      e,' '
          call    serout
; now determine free space available
          call    txenq
          ld      e,'F'
          call    dskout
          call    term
          call    getackn
          jr      c,dskf
          and     a
          jr      nz,dskf
          call    ack
          call    serin
          jr      c,dskf
          ld      l,a
          call    serin
          jr      c,dskf
          ld      h,a
          jr      dsko
;
dskf:     ld      hl,0            ;if error output zero spcae
;
dsko:     sla     l
          rl      h
          ld      de,poolarea
          call    cns
          ld      hl,bytex
          ld      bc,17
          ldir
          ld      a,'$'
          ld      (de),a
          call    linehl
ok:       xor     a
          ret

bytex:    db      'K bytes available'
;
; search for first extent of file
;
search1:  call    txenq
          ld      e,'S'
          call    issue
ser1:     ret     c               ;if timeout branch
          and     a
          ret     nz
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
; a file has been found
          call    ack
          ld      hl,dma
          ld      b,32
kerd1:    call    serin
          ret     c               ;if timeout return
          ld      (hl),a
          inc     hl
          djnz    kerd1
          ld      hl,(dma+10h)
          ld      (tempts),hl     ;keep position
          xor     a
          ret


; search for next occurence of extent
search2:  push    hl
          call    txenq
          ld      e,'N'
          call    dskout
          pop     hl
          inc     hl
          ld      b,12
          call    string
          ld      de,(tempts)
          push    de
          ld      e,d
          call    dskout
          pop     de
          call    dskout
          call    term
          call    getackn
          jr      ser1
; routine to check if a disk is read only or not


tstdskro: call    txenq
          ld      e,'Q'
          call    dskout
          call    term
          call    serin           ;get disk status
          push    af
          call    serin
          jr      nc,tsd1
; timeout occured so set carry on return
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited

Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
          pop     af
          scf
          ret
tsd1:     pop     af
          and     20h             ;test read bi
          ret

SCRNDIS:  LD      B,8
          CALL    SCRNDIS1        ;move in first part of name
          LD      A,'.'
          LD      (DE),A          ;put in '.' seperator
          INC     DE
          LD      B,3             ;put out terminator
SCRNDIS1: LD      A,(HL)
          AND     7FH             ;strip of attributes
          LD      (DE),A
          INC     DE
          INC     HL
          DJNZ    SCRNDIS1
          RET

;wildcard check routine

wldchk:   ld      a,'?'
          ld      b,11
wld1:     cp      (hl)
          ret     z
          inc     hl
          djnz    wld1
          ld      a,1
          and     a
          ret
AXHL:     PUSH    DE              ;SAVE DE
          EX      DE,HL
          LD      HL,0            ;ZERO HL FOR RESULT
          AND     A               ;CHECK FOR MULTIPLY BY 0
          JR      Z,AXHL1         ;IF SO THEN FINISHED
AXHL2:    ADD     HL,DE
          DEC     A
          JR      NZ,AXHL2        ;LOOP ROUND TIL DONE
;
AXHL1:    POP     DE              ;RETRIEVE DE
          RET
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
;
;               CONVERT INTEGER TO STRING
;       DE CONTAINS ADDR OF STRING, RETURN UPDATED VALUE IN DE
;       HL CONTAINS VALUE TO CONVERT
;
CNS:      XOR     A               ;SET FOR NO LEADING ZEROES
          LD      BC,-10000
          CALL    RSUB
          LD      BC,-1000
          CALL    RSUB
          LD      BC,-100
          CALL    RSUB
          LD      BC,-10
          CALL    RSUB
          LD      BC,-1
          CALL    RSUB
          RET     NZ
          LD      A,'0'
          LD      (DE),A
          INC     DE
          RET
;
;       TAKE VALUE IN HL
;       SUB MINUS NUMBER IN BE THE MOST POSSIBLE TIMES
;       PUT VALUE ON STRING AT DE
;       IF A=0 THEN DONT PUT ZERO ON STRING
;       RETURN NON-ZERO IN A IF PUT ON STRING
;
RSUB:     PUSH    DE
          LD      D,0FFH
RSUB1:    PUSH    HL
          POP     IX
          INC     D
          ADD     HL,BC
          JR      C,RSUB1
          PUSH    IX
          POP     HL
          LD      B,D
          POP     DE
          OR      B               ;A GETS 0 IF A WAS 0 AND B IS 0
          RET     Z
          LD      A,'0'
          ADD     A,B
          LD      (DE),A
          INC     DE
          RET
;
```

The information contained in this document is ©opyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damages resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
PRNLNE:   LD      HL,POOLAREA     ;clear buffer
          LD      B,40
PRNLNY:   LD      (HL),' '
          INC     HL
          DJNZ    PRNLNY
;
          LD      HL,(WORKR11)
          PUSH    HL              ;save EX and CR
          EX      DE,HL           ;save them in DE
          LD      L,D
          LD      H,0             ;HL = EX
          LD      A,128
          CALL    AXHL            ;get no. of recs in extents
          LD      D,0
          ADD     HL,DE           ;get total no. records in HL
          PUSH    HL              ;save no. of records
          LD      DE,POOLAREA     ;initialize pointer
          LD      A,255           ;select leading zeros
          PUSH    DE              ;save pointer
          CALL    CNS+1           ;put no. in pool as string
          EX      DE,HL           ;put new pointer in HL
;
          EX      (SP),HL         ;get old pointer and save new one
          CALL    ZEROKL          ;blank out the leading zeros
          POP     DE              ;get new pointer
          LD      A,' '
          LD      (DE),A          ;insert space
          INC     DE              ;advanc to next field
          EX      DE,HL
          EX      (SP),HL         ; get no. of records and save pointer
          CALL    KCONVRT         ;get size in Ks
          JR      Z,SKIPZA        ;if exact then dont increment
          INC     HL
SKIPZA:   BIT     0,L             ;if an odd value inc again
          JR      Z,SKIPZ
          INC     HL              ;round up to the next no.of K if odd
SKIPZ:    POP     DE              ;get pointer
;
          PUSH    DE              ;save it
          LD      A,255           ;select leading zeros
          CALL    CNS+1           ;put in length in pool as string
          EX      DE,HL           ;put pointer in HL
          EX      (SP),HL         ;get old pointer and save new one
          CALL    ZEROKL          ;blank out the leading zeros
          POP     DE              ;get new pointer
          LD      A,'K'
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
            LD      (DE),A          ;insert 'K' suffix
            INC     DE              ;point to next field
;
            POP     HL              ;get no. of extents
            PUSH    DE              ;save pointer
            LD      L,H
            LD      H,0
            INC     HL              ;HL = no. of extents of the file
            LD      A,255           ;select leading zeros
            CALL    CNS+1           ;put extents in pool as a string
            EX      DE,HL           ;put pointer in HL
            EX      (SP),HL         ;get old pointer and save new one
            CALL    ZEROKL          ;blank out the leading zeros
;
            POP     HL
            LD      (HL),' '
            INC     HL
            LD      (HL),'R'
            INC     HL
            LD      (HL),'/'
            INC     HL
            LD      (HL),'W'
            PUSH    HL
            LD      HL,DMA+9
            BIT     7,(HL)          ;test if r/o file
            POP     HL
            JR      Z,PRNLN1
            LD      (HL),'O'
PRNLN1:     INC     HL
            LD      (HL),' '
            INC     HL
            LD      (HL),':'
            INC     HL
            EX      DE,HL
            LD      HL,DMA+1
            PUSH    HL
            POP     IX
            BIT     7,(IX+9)        ;test for system file
            JR      Z,PRNLIN2
            LD      A,'('           ;insert open bracket
            LD      (DE),A
            INC     DE
PRNLIN2:    CALL    SCRNDIS
            BIT     7,(IX+9)
            JR      Z,PRNLIN4
            LD      A,')'           ;insert close bracket
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder. Husky Computers Limited



Disclaimer. Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
            LD      (DE),A
PRNLIN4:    LD      A,'$'
            LD      (POOLAREA+40),A;put in the terminator
LINEHL:     LD      DE,POOLAREA
            XOR     A
LINEOUT:    JP      PRN1
;
ZEROKL:     LD      B,4
ZEROLP:     LD      A,(HL)          ;get character
            CP      '0'
            RET     NZ              ;finished if not a 0
            LD      (HL),' '        ;blank out with space
            INC     HL              ;point to next character
            DJNZ    ZEROLP          ;maximun of 4 times
            RET
;
SERINC:     PUSH    HL              ;keep regs
            PUSH    DE
            PUSH    BC
SERINC1:    CALL    DSKSTAT         ;any character yet
            AND     A
            JP      NZ,SERIN2       ;if yes branch
            LD      C,11
            CALL    SYSTEM          ;check for keyboard activity
            AND     A
            JR      Z,SERINC1       ;look for next
            LD      C,1
            CALL    SYSTEM          ;fetch the key
            CP      3
            JR      NZ,SERINC1      ;if not control c then loop
; control C has been detected
            POP     BC              ;restore regs
            POP     DE
            POP     HL
            LD      A,-1
            SCF                     ;indicate error and return
            RET
REPLYQ:     CALL    SERINC
            JP      C,GETACKF       ;if abort branch
            CALL    CONVH
            RRCA
            RRCA
            RRCA
            RRCA
            LD      B,A
            CALL    SERINC          ;get termintaor
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder. Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
            JP      C,GETACKF       ;if timeout branch
            CALL    CONVH
            ADD     A,B
            LD      B,A             ;keep char
            CALL    SERINC
            CP      CR              ;check if cr
            JP      NZ,GETACKF      ;if not branch
            LD      A,B
            AND     A
            RET
;
;
;    <<<<<< Convert 128 byte size in HL into size in K >>>>>>>
;
KCONVRT:    LD      A,L
            AND     07H             ;check for odd no. of records not K
            PUSH    AF
            SRL     H
            RR      L               ;divide by two
            SRL     H
            RR      L               ;divide by four
            SRL     H
            RR      L               ;divide by eight:HL=leng of file in K
            POP     AF              ;
            RET

header:     db      ' Recs  Bytes  Ext Acc',cr,lf,'$'

;*****************************************
; Routine to calculate file size of a file
;*****************************************

filesize:   ld      hl,(tempts)     ;preserve position
            ld      (temptsl),hl

            ld      hl,dma          ;transfer name to fcb
            ld      de,tbuf
            ld      bc,32
            ldir                    ;copy filename to tbuf

            ld      ix,tbuf
            ld      a,(ix+12)       ;fetch parameters
            ld      (exfield),a
            ld      a,(ix+15)
            ld      (rcfield),a
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.



Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
; now continue search for other extents
zcfb:     ld      a,'?'
          ld      (tbuf+12),a     ;set extent wildcard
zcfa:     ld      hl,tbuf
          call    search2         ;look for next entry
          ret     c               ;if timeout branch
          and     a
          jr      nz,zcfr         ;if no more branch
; extent found
          ld      ix,dma
          ld      a,(exfield)     ;compare parameters
          cp      (ix+12)
          jr      nc,zcfa         ;last one larger so ignore this
; new one is larger
          ld      a,(ix+12)       ;fetch parameters
          ld      (exfield),a
          ld      a,(ix+15)
          ld      (rcfield),a
          jr      zcfb            ;loop again
; largest extent found so calculate file size
zcfr:     ld      a,(exfield)
          ld      h,a
          ld      a,(rcfield)
zcfr1:    ld      l,a
; hl now contains the number of records in the file
          push    hl
          ld      hl,(tempts1)
          ld      (tempts),hl
          pop     hl
          and     a               ;clear error flag
          ret


$eject
;====================================
; DISK LOAD FUNCTION
;====================================


;           DE = fcb containing the filename (wildcards allowed)


kerload:  call    initfcb         ;initialise the fcb work area
; copy name to tbuf1
          call    move
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited



Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
; look for the file on the Oracle GT
          xor     a
          ld      (tbuf1+12),a    ;look for first extent
          call    search1
          jp      c,dferr3
          and     a
          jp      nz,dferr1       ;if no  file branch
; need to perform the disk size calculation
kerlp:    call    filesize        ;no of records in workr11
          jp      c,dferr3
; calculate the no of records
          ex      de,hl
          ld      l,d
          ld      h,0
          ld      a,128
          call    axhl
          ld      d,0
          add     hl,de
          ld      (workr11),hl
;check keyboard for abort control C
load1:    call    keyst
          and     a
          jr      z,load1a
          call    keyin
          cp      3
          jr      nz,load1
          jp      dferr6          ;if control C abort
; now copy filename to the working fcb
load1a:   ld      hl,dma
          ld      de,tbuf
          ld      bc,13
          ldir
; delete the file on the ram disk
          xor     a
          ld      (tbuf+12),a     ;clear the extent byte
          ld      de,tbuf
          call    delete
; now make the file
          ld      de,tbuf
          call    make
          inc     a
          jp      z,dferr5        ;if error branch
; now check if progress is to be displayed
          call   opfile1          ;if allowed output filename
; commence transfer of the data from file to file
          ld      hl,0
loadx:    ld      (sectcnt),hl    ;load start sector count
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited



Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited PO Box 436, 345 Foleshill Road, Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
; read the disk sector from Oracle GT
          call    txenq
          ld      e,'R'
          call    dskout
          ld      a,(tempts)
          ld      e,a
          call    dskout
          ld      de,(sectcnt)
          push    de
          call    dskout
          pop     de
          ld      e,d
          call    dskout
          call    term
          call    getackn
          jp      c,dferr3        ;if timeout branch
          and     a
          jp      nz,dferr3       ;if failed then error
; data waitinf to be sent
          call    ack             ;issue acknowledge
          ld      hl,dma
          ld      c,0             ;init checksum
          ld      b,128
load3:    call    serin           ;fetch data
          jp      c,dferr3        ;if timeout branch
          ld      (hl),a
          add     a,c             ;accumulate checksum
          ld      c,a
          inc     hl
          djnz    load3
          call    serin           ;get checksum
          jp      c,dferr3
          add     a,c
          inc     a
          jp      nz,dferr3       ;if bad checksum give disk error
; data record is now in the dma buffer
          ld      de,tbuf
          call    writes
          and     a
          jp      nz,dferr7       ;issue failed message
; increment to next record
          ld      hl,(sectcnt)
          inc     hl
          ld      de,(workrll)
          and     a
          sbc     hl,de
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited



Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
          add     hl,de
          jp      nz,loadx        ;loop to for next record
; read is completed
load4:    ld      de,tbuf
          call    close           ;close file on ram disk
;
; need to check if more files area available
          call    crlf            ;issue line terminator
; issue the search next function to Oracle GT
          xor     a
          ld      (tbuf1+12),a
          ld      hl,tbuf1
          call    search2
          jp      c,dferr3        ;if timeout then branch
          and     a
          jp      z,kerlp         ;if more branch
; no further file is available so loop for next
          xor     a
          ret


$eject
;====================================
; DISK DUMP FUNCTION
;====================================


; on entry require the filename

kerdump:  call    initfcb         ;initialise the fcb work area
          ld      de,tbuf
          call    searchf         ;look for occurence of file
          inc     a
          jp      z,dferr1        ;if file not found branch
; ok file has been found
dump1:    ld      hl,(lstdir)     ;fetch track/sector info
          ld      (tempts),hl     ;store away
          ld      hl,(lstpge)
          ld      (tempge),hl
; check keyboard for break condition
keyloop:  call    keyst           ;test keyboard status
          and     a
          jr      z,dumpla        ;if nothing continue
          call    keyin           ;fetch the key
          cp      3               ;is it control c
          jr      nz,keyloop
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited



Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
; break condition detected
          jp      dferr6          ;goto break error
; copy filename
dumpla:   ld      hl,dma
          call    movex
; test if screen update is required
          call    opfile          ;output filename if allowed
; now perform the transfer
dump2:    call    tstdskro        ;test if disk is read only or not
          jp      c,dferr3        ;if timeout branch
          jp      nz,dferr9       ;if yea then branch
          xor     a
          ld      (tbuf1+12),a    ;clear ex
          ld      (tbuf1+32),a    ;clear cr
          ld      de,tbuf1
          call    open            ;open the source file
          inc     a
          jp      z,dferr2
; now delete the file on the oracle GT
          call    txenq
          ld      e,'D'
          call    issue
          jp      c,dferr3        ;if timeout branch
          cp      0A7H
          jp      z,dferr4        ;if read only file branch
; now make the new file
          call    txenq
          ld      e,'M'
          call    issue           ;make the file
          jp      c,dferr3        ;if timeout branch
          and     a
          jp      nz,dferr5       ;if failed to make file then error
          call    ack             ;issue acknowledge
          call    serin           ;get reply
          jp      c,dferr3        ;if timeout branch
          ld      l,a
          ld      h,0
          ld      (dskparm),hl    ;store the disk parameters
; files are now open and ready transfer
          ld      hl,0
dump3:    ld      (sectcnt),hl    ;initialise sector counter
          ld      de,tbuf1
          call    reads           ;perform file read
          and     a
          jp      nz,dump4        ;if nothing assume file end
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder. Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
; now write the sector to the disk
          call    txenq
          ld      e,'W'
          call    dskout
          ld      a,(dskparm)
          ld      e,a
          call    dskout
          ld      de,(sectcnt)    ;fecth sector counter
          push    de
          call    dskout
          pop     de
          ld      e,d
          call    dskout
          call    term
          call    getackn
          jp      c,dferr3
          and     a
          jp      nz,dferr5
; tx the dma buffer
          ld      hl,dma
          ld      b,80h
          xor     a               ;clear checksum accumulator
wloop:    ld      e,(hl)          ;get char
          add     a,e             ;form checksum
          push    af
          push    bc
          push    hl
          call    dskout
          pop     hl
          pop     bc
          pop     af
          inc     hl
          djnz    wloop           ;look if more chars
          cpl
          ld      e,a
          call    dskout          ;output checksum
          ld      hl,(sectcnt)
          inc     hl              ;increment sector
; read completed ok so loop to next sector
          jp      dump3
; transfer completed
dump4:    call    txenq
          ld      e,'C'
          call    dskout          ;now close file on disk
          call    term
          call    getackn         ;fecth reply
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder. Husky Computers Limited

Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
; issue crlf if outputting to screen
          call    crlf
; look for other file
          ld      hl,(tempts)     ;fetch track sector info
          ld      (lstdir),hl
          ld      hl,(tempge)
          ld      (lstpge),hl
; need to update the search fcb again
          ld      hl,tbuf
          ld      de,0E05CH
          ld      bc,13
          ldir
          call    searchn
          inc     a
          ret     z               ;if no more files return
; loop for next file
          jp      dump1


$eject
;=====================================
; DISK ERASE FUNCTION
;=====================================

; on entry: A  = non zero if wand echo to screen else no echo
;           DE = fcb containing the filename (wildcards allowed)


kerera:   call    initfcb
; test for wildcards in name
          ld      hl,tbuf+1
          call    wldchk
          jp      z,dferr8        ;if wildcards then error
          call    move            ;translate data to other fcb
          call    tstdskro        ;test if disk is readonly or not
          jp      c,dferr3        ;if timeout branch
          jp      nz,dferr9       ;if yes then branch
          call    txenq
          ld      e,'D'
          call    issue           ;send command and get ack
          jp      c,dferr3        ;if timeout branch
          cp      0A7H
          jp      z,dferr4        ;if read only branch
          jp      ok
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder. Husky Computers Limited



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
$eject
;====================================
; DISK RENAME FUNCTION
;====================================

; on entry: A  = non zero if wand echo to screen else no echo
;           DE = fcb containing the filename (wildcards allowed)


kerren:   call    initfcb
          ld      hl,(stack)
          ld      a,(hl)
          inc     hl
          ld      h,(hl)
          ld      l,a
          ld      de,12
          add     hl,de
          ld      de,tbuf+17
          ld      bc,11
          ldir
; test for wildcards in both names
          ld      hl,tbuf+1
          call    wldchk
          jp      z,dferr8
          ld      hl,tbuf+17
          call    wldchk
          jp      z,dferr8
          call    tstdskro        ;test dsk is set to read only or not
          jp      c,dferr3        ;if timeout branch
          jp      nz,dferr9       ;if yes then branch
          call    txenq
          ld      e,'L'           ;issue command
          call    dskout
          ld      hl,tbuf+1       ;output the filenames
          ld      b,31
kerren1:  ld      e,(hl)
          push    hl
          push    bc
          call    dskout
          pop     bc
          pop     hl
          inc     hl
          djnz    kerren1
          call    term
          call    getackn         ;fetch reply
          ret
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181. Telex: 317450 Husky G.

```
$eject
;====================================
; DISK FORMAT FUNCTION
;====================================

kerform:  push    bc              ;keep size
          call    txenq
          ld      e,'X'
          call    dskout          ;issue format label
          pop     af
          ld      e,a
          call    dskout          ;issue disk density
          call    term
; now await reply
          call    replyq          ;get reply
          jr      c,kerf1         ;if bad then branch
          and     a
          ret     z               ;if ok return
kerf1:    scf
          ret


$eject
;=============================================================================
;
;
;                COMMON ROUTINES REQUIRED BY ALL FUNCTIONS
;
;
;=============================================================================

; CP/M routines
;

searchf:  ld      c,17
          jr      sys
searchn:  ld      c,18
          jr      sys
open:     ld      c,15
          jr      sys
reads:    ld      c,20
          jr      sys
dskout:   ld      c,4
          jr      sys
keyst:    ld      c,11
          jr      sys
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited

Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
keyin:     ld      c,47          ;special no echo to screen call
           jr      sys
serout:    ld      c,2
           jr      sys
dskin:     ld      c,3
           jr      sys
dskst:     ld      c,42
           jr      sys
delete:    ld      c,19
           jr      sys
make:      ld      c,22
           jr      sys
writes:    ld      c,21
           jr      sys
close:     ld      c,16
           jr      sys
comsetup:  ld      c,26h
           jr      sys
prnstrg:   ld      c,9
;
sys:       push    hl
           call    5h
           pop     hl
           ret

;==================================
; disk access routines
;==================================


ENQ:       EQU     05H

TIME:      EQU     4000H
TIM1:      EQU     1


ISSUE:     CALL    DSKOUT
           LD      B,12
           LD      HL,TBUF1+1    ;output filename
           CALL    STRING
           CALL    TERM          ;output acknowledge
;
; Routine to receive acknowledge
;
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
GETACKN:  CALL    SERIN
          JR      C,GETACKF       ;if timeout branch
          CALL    CONVH
          RRCA
          RRCA
          RRCA
          RRCA
          LD      B,A
          CALL    SERIN           ;get termintaor
          JR      C,GETACKF       ;if timeout branch
          CALL    CONVH
          ADD     A,B
          LD      B,A             ;keep char
          CALL    SERIN
          CP      CR              ;check if cr
          JR      NZ,GETACKF      ;if not branch
          LD      A,B
          AND     A
          RET

STRING:   LD      E,(HL)          ;get char
          PUSH    HL
          PUSH    BC
          CALL    DSKOUT          ;output it
          POP     BC
          POP     HL
          INC     HL
          DJNZ    STRING          ;any more
          RET
;
TXENQ:    LD      E,ENQ
          JP      DSKOUT
;
TERM:     NOP
ACK:      LD      E,CR            ;issue acknowledge
          JP      DSKOUT
;
; Routine to fetch a character
;
SERIN:    PUSH    HL              ;keep regs
          PUSH    DE
          PUSH    BC
          LD      HL,TIME         ;load timeout values
          LD      B,TIM1
SERIN1:   CALL    DSKSTAT         ;any character yet
          AND     A
          JR      NZ,SERIN2       ;if yes branch
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited

Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damages resulting from the use of the information contained within this document



Husky Computers Limited PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
; no character so dec timers
            DEC     HL
            LD      A,L
            OR      H
            JR      NZ,SERIN1
            DJNZ    SERIN1
; timeout has occured
            POP     BC              ;restore regs
            POP     DE
            POP     HL
GETACKF:    LD      A,-1
            SCF                     ;indicate error and return
            RET
; character available
SERIN2:     CALL    DSKSERIN
            POP     BC
            POP     DE
            POP     HL
            AND     A
            RET
;
;
;
CONVH:      CP      3AH
            JR      C,CONVH1
            SUB     37H
CONVH1:     AND     0FH
            RET
;
;
DSKSTAT:    PUSH    HL
            PUSH    BC
            PUSH    DE
            CALL    DSKST
            JR      DSKXIT
;
DSKSERIN:   PUSH    HL
            PUSH    BC
            PUSH    DE
            CALL    DSKIN
DSKXIT:     POP     DE
            POP     BC
            POP     HL
            RET
CALCSECT:   LD      DE,80H
            LD      A,(FCBEX)       ;fetch extent field
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder Husky Computers Limited



Disclaimer Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document

Husky Computers Limited PO Box 135 345 Foleshill Road Coventry CV6 5RW England Telephone Coventry (0203) 668181 Telex 317450 Husky G

```
            LD      B,A
            LD      HL,0            ;init accumulator
            AND     A
            JR      Z,CALCS2
CALCS1:     ADD     HL,DE
            DJNZ    CALCS1
; determined offset due to extent in HL
CALCS2:     LD      A,(TBUF1+32)    ;add in offset record
            LD      E,A
            LD      D,0
            ADD     HL,DE
; HL now contains the record required from the file
            RET


SCRNOUT:    PUSH    BC              ;output string to screen
            LD      A,(HL)
            AND     7FH             ;remove attribute bits
            LD      E,A
            CALL    SEROUT
            POP     BC
            INC     HL
            DJNZ    SCRNOUT
            RET

INITFCB:    LD      HL,TBUF         ;initialise the buffer
            LD      DE,TBUF+1
            LD      BC,35
            LD      (HL),0
            LDIR
            LD      HL,(STACK)
            LD      A,(HL)
            INC     HL
            LD      H,(HL)
            LD      L,A
            LD      DE,TBUF+1       ;copy name to buffer
            LD      BC,12
            LDIR
            XOR     A
            LD      (TBUF),A        ;ensure dr field is zero
            LD      (TBUF+12),A     ;ensure ex field is zero
            RET
;
; move fcb routine
;
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
move:     ld      hl,tbuf
movex:    ld      de,tbuf1
          ld      bc,13
          ldir
          ret

; display progress on the screen of the copy

opfile1:  ld      hl,tbuf+1
          jr      opf1
opfile:   ld      hl,tbuf1+1
opf1:     ld      b,8
          call    scrnout
          ld      e,'.'
          call    serout
          ld      b,3
          call    scrnout
          ld      e,' '
          jp      serout
;
; set comms parameters
;
combuf:   equ     0F7F8H
serig:    equ     0f84BH
timeouts: equ     0F85DH

setcom:   ld      hl,0
          ld      (timeouts),hl  ;clear comms timeouts
          ld      de,combuf
          ld      hl,comparms
          ld      bc,0EH
          ldir                   ;transfer parameters
          ld      a,80h
          ld      (serig),a      ;turn off serig
          jp      comsetup       ;perform comms setup call

comparms: db      9              ;tx = 4800
          db      9              ;rx = 4800
          db      1              ;cts = no
          db      1              ;dtr = yes
          db      1              ;rts = hold
          db      0              ;dsr = no
          db      0              ;dcd = no
          db      3              ;tx pty = 8
          db      3              ;rx pty = 8
          db      0              ;tx protocol none
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damages resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668161. Telex: 31450 Husky G.

```
            db      0               ;lf no
            db      0               ;nulls = 0
            db      0               ;echo no
            db      0               ;rx protocol none

;============================
; error exit points
;============================

dferr1:     ld      de,msg1
            jr      prn
dferr2:     ld      de,msg2
            jr      prn
dferr3:     ld      de,msg3
            jr      prn
dferr4:     ld      de,msg4
            jr      prn
dferr5:     ld      de,msg5
            jr      prn
dferr6:     ld      de,msg6
            jr      prn
dferr7:     ld      de,msg7
            jr      prn
dferr8:     ld      de,msg8
            jr      prn
dferr9:     ld      de,msg9
;
; output the error string terminated by line characters
;
prn:        ld      a,(de)
            inc     de
prn1:       push    af              ;store error number
            ld      c,9
            call    system          ;output error message
            call    crlf
            pop     af              ;restore error number
            ret
;
crlf:       ld      e,cr
            ld      c,2
            call    system
            ld      e,lf
            ld      c,2
            jp      system
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document, the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

```
msg1:     db      0B1h,'- no file$'
msg2:     db      0B2h,'- cannot open file$'
msg3:     db      0B3h,'- disk error$'
msg4:     db      0B4h,'- file read only$'
msg5:     db      0B5h,'- cannot make file$'
msg6:     db      0B6h,'- aborted$'
msg7:     db      0B7h,'- cannot write to file$'
msg8:     db      0B8h,'- filename error$'
msg9:     db      0B9h,'- disk read only$'

end:      nop                     ;very last byte of code

; ram definitions

tempts:   ds      2               ;track and sector store
tempge:   ds      2               ;last page access
dskparm:  ds      2               ;disk info returned by Oracle GT
stem:     ds      2               ;buffer pointer
len:      ds      1               ;length remaining in buffer
sectcnt:  ds      2               ;sector counter register

density:  ds      1               ;disk density

exfield:  ds      1               ;extent register
rcfield:  ds      1               ;record register
workrll:  ds      2               ;temp storage

tempts1:  ds      2               ;secondary buffer

stack:    ds      2               ;user stack pointer

poolarea: ds      40              ;temporary line buffer

tbuf:     ds      35
tbuf1:    ds      35

          END
```

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical or photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damages resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 335, 335 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

## 7. COMMUNICATION DETAILS OF INTERFACE

This mode of operation provides a means of communication between a system file executed on a CP/M computer and a disk configured in a compatible IBM PC format.

All commands in this mode are prefixed by an ENQ character (05H).

Specifications - RS232 full duplex

Capacity - IBM PC compatible format
720K bytes
1.4M bytes

Baud rate - Software selectable
150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400. However, for use with Husky Hunter the baud rate should be set at 4800 baud.

Character Format - One start bit, 8 data bits, 1 stop bit

The command protocol consists of the following segments;

1. **REQUEST BLOCK:** A fixed form dependent on the command, sent from the computer to the drive unit.

2. **ACKNOWLEDGE BLOCK:** A hedadecimal reply send as two ascii bytes terminated by carriage return.

3. **REQUEST DATA:** An optional request dependent on the command for requested information.

4. **REQUEST DATA:** Requested data sent from the drive unit to the computer, the format of which is dependent on the original command.

The information contained in this document is copyright and may not be reproduced, stored in a retrieval system in any form or by any means, electronic, mechanical, photocopying or otherwise, without prior permission from the copyright holder, Husky Computers Limited.



Disclaimer: Whilst every precaution has been taken in the preparation of this document the publisher assumes no responsibility for errors and omissions, neither is any liability assumed for damage resulting from the use of the information contained within this document.

Husky Computers Limited, PO Box 135, 345 Foleshill Road, Coventry CV6 5RW England. Telephone: Coventry (0203) 668181 Telex: 317450 Husky G

SEARCHF - Search directory for first occurrence of a specified filename.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | | | ENQ | | | | | |
| | 2 | Ø | 1 | Ø | 1 | Ø | Ø | 1 | 1 |
| | 3 | | | F | | | | | |
| | 4 | | | I | | | | | |
| | 5 | | | L | | | | | |
| | 6 | | | E | | | | | |
| | 7 | | | N | | | | | |
| | 8 | | | A | | | | | |
| | 9 | | | M | | | | | |
| | 10 | | | E | | | | | |
| | 11 | | | E | | | | | |
| | 12 | | | X | | | | | |
| | 13 | | | T | | | | | |
| | 14 | | | EX | | | | | |
| | 15 | | | CR | | | | | |
| ACKNOWLEDGE BLOCK | 1 | | | ACKH | | | | | |
| | 2 | | | ACKL | | | | | |
| | 3 | | | CR | | | | | |
| REQUEST DATA | 1 | | | CR | | | | | |
| RETURN DATA | 1 | | | Ø | | | | | |
| | 2 | | | F | | | | | |
| | 3 | | | I | | | | | |
| | 4 | | | L | | | | | |
| | 5 | | | E | | | | | |
| | 6 | | | N | | | | | |
| | 7 | | | A | | | | | |
| | 8 | | | M | | | | | |
| | 9 | | | F | | | | | |
| | 10 | | | E | | | | | |
| | 11 | | | Y | | | | | |
| | 12 | | | T | | | | | |
| | 13 | | | Ø | | | | | |
| | 14 | | | | | | | | |
| | 15 | | | | | | | | |
| | 16 | | | IBM PARAM | | | | | |
| | 17 | | | CPM PARAM | | | | | |
| | 31 | | | | | | | | |

The two parameters IBM PARAM and CPM PARAM should be preserved and used for subsequent SEARCHN commands. The filename returned in the return data block corresponds to the matching directory entry for the name used in the search command.

**SEARCHN** - Search directory for next occurrence of a specified filename. This command may only be used after a SEARCHF command.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | | | ENQ | | | | | |
| | 2 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| | 3 | | | F | | | | | |
| | 4 | | | T | | | | | |
| | 5 | | | L | | | | | |
| | 6 | | | E | | | | | |
| | 7 | | | N | | | | | |
| | 8 | | | A | | | | | |
| | 9 | | | M | | | | | |
| | 10 | | | E | | | | | |
| | 11 | | | E | | | | | |
| | 12 | | | X | | | | | |
| | 13 | | | T | | | | | |
| | 14 | | | EX | | | | | |
| | 15 | | | CR | | | | | |
| ACKNOWLEDGE BLOCK | 1 | | | ACKH | | | | | |
| | 2 | | | ACKL | | | | | |
| | 3 | | | CR | | | | | |
| REQUEST DATA | 1 | | | CR | | | | | |
| RETURN DATA | 1 | | | Ø | | | | | |
| | 2 | | | F | | | | | |
| | 3 | | | I | | | | | |
| | 4 | | | L | | | | | |
| | 5 | | | E | | | | | |
| | 6 | | | N | | | | | |
| | 7 | | | A | | | | | |
| | 8 | | | M | | | | | |
| | 9 | | | E | | | | | |
| | 10 | | | E | | | | | |
| | 11 | | | X | | | | | |
| | 12 | | | T | | | | | |
| | 13 | | | Ø | | | | | |
| | 14 | | | | | | | | |
| | 15 | | | | | | | | |
| | 16 | | | IBM PARAM | | | | | |
| | 17 | | | CPM PARAM | | | | | |
| | 31 | | | | | | | | |

The two parameters IBM PARAM and CPM PARAM should be preserved and used for subsequent SEARCHN commands. The filename returned in the return data block corresponds to the matching directory entry for the name used in the SEARCHF command.

**OPEN** Search directory for first occurrence of a specified filename in preparation for read write access.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | ENQ | | | | | | | |
| | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| | 3 | F | | | | | | | |
| | 4 | T | | | | | | | |
| | 5 | L | | | | | | | |
| | 6 | E | | | | | | | |
| | 7 | N | | | | | | | |
| | 8 | A | | | | | | | |
| | 9 | M | | | | | | | |
| | 10 | E | | | | | | | |
| | 11 | E | | | | | | | |
| | 12 | X | | | | | | | |
| | 13 | T | | | | | | | |
| | 14 | EX | | | | | | | |
| | 15 | CR | | | | | | | |
| ACKNOWLEDGE BLOCK | 1 | ACKH | | | | | | | |
| | 2 | ACKL | | | | | | | |
| | 3 | CR | | | | | | | |
| REQUEST DATA | 1 | CR | | | | | | | |
| RETURN DATA | 1 | 0 | | | | | | | |
| | 2 | F | | | | | | | |
| | 3 | I | | | | | | | |
| | 4 | L | | | | | | | |
| | 5 | E | | | | | | | |
| | 6 | N | | | | | | | |
| | 7 | A | | | | | | | |
| | 8 | M | | | | | | | |
| | 9 | F | | | | | | | |
| | 10 | E | | | | | | | |
| | 11 | X | | | | | | | |
| | 12 | T | | | | | | | |
| | 13 | EX | | | | | | | |
| | 14 | | | | | | | | |
| | 15 | | | | | | | | |
| | 16 | TRM PARAM | | | | | | | |
| | 17 | CPM PARAM | | | | | | | |
| | 31 | | | | | | | | |

The returned file control block should be preserved for subsequent access.

**DELETE** - Search directory for occurrence of a specified explicit filename and delete all entries.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | | | ENQ | | | | | |
| | 2 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 3 | | | F | | | | | |
| | 4 | | | I | | | | | |
| | 5 | | | L | | | | | |
| | 6 | | | E | | | | | |
| | 7 | | | N | | | | | |
| | 8 | | | A | | | | | |
| | 9 | | | M | | | | | |
| | 10 | | | E | | | | | |
| | 11 | | | E | | | | | |
| | 12 | | | X | | | | | |
| | 13 | | | T | | | | | |
| | 14 | | | EX | | | | | |
| | 15 | | | CR | | | | | |
| ACKNOWLEDGE BLOCK | 1 | | | ACKH | | | | | |
| | 2 | | | ACKL | | | | | |
| | 3 | | | CR | | | | | |

The drive unit searches the directory for the occurrence of the specified filename and deletes the entry if found, provided the file is not protected.

**RENAME** - Search directory for occurrence of a specified explicit filename and change the name of the file if found, to another specified name.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | | | ENQ | | | | | |
| | 2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| | 3 | | | F | | | | | |
| | 4 | | | I | | | | | |
| | 5 | | | L | | | | | |
| | 6 | | | E | | | | | |
| | 7 | | | N | | | | | |
| | 8 | | | A | | | | | |
| | 9 | | | M | | | | | |
| | 10 | | | E | | | | | |
| | 11 | | | E | | | | | |
| | 12 | | | X | | | | | |
| | 13 | | | T | | | | | |
| | 14 | | | 0 | | | | | |
| | 15 | | | 0 | | | | | |
| | 16 | | | 0 | | | | | |
| | 17 | | | 0 | | | | | |
| | 18 | | | N | | | | | |
| | 19 | | | E | | | | | |
| | 20 | | | W | | | | | |
| | 21 | | | N | | | | | |
| | 22 | | | A | | | | | |
| | 23 | | | M | | | | | |
| | 24 | | | E | | | | | |
| | 25 | | | | | | | | |
| | 26 | | | E | | | | | |
| | 27 | | | X | | | | | |
| | 28 | | | T | | | | | |
| | 29 | | | 0 | | | | | |
| | 30 | | | 0 | | | | | |
| | 31 | | | 0 | | | | | |
| | 32 | | | 0 | | | | | |
| | 33 | | | 0 | | | | | |

| | | |
|---|---|---|
| ACKNOWLEDGE BLOCK | 1 | ACKH |
| | 2 | ACKL |
| | 3 | CR |

Bytes 3 - 13 contain the filename to be altered.
Bytes 18 - 28 contain the new filename.

**MAKE** - Create a directory entry on the disk. The user must ensure a similar filename does not already exist.

| | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | | | | ENQ | | | | | |
| | 2 | | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| | 3 | | | | F | | | | | |
| | 4 | | | | I | | | | | |
| | 5 | | | | L | | | | | |
| | 6 | | | | E | | | | | |
| | 7 | | | | N | | | | | |
| | 8 | | | | A | | | | | |
| | 9 | | | | M | | | | | |
| | 10 | | | | E | | | | | |
| | 11 | | | | E | | | | | |
| | 12 | | | | X | | | | | |
| | 13 | | | | T | | | | | |
| | 14 | | | | EX | | | | | |
| | 15 | | | | CR | | | | | |
| ACKNOWLEDGE BLOCK | 1 | | | | ACKL | | | | | |
| | 2 | | | | ACKH | | | | | |
| | 3 | | | | CR | | | | | |
| REQUEST DATA | 1 | | | | CR | | | | | |
| RETURN DATA | 1 | | | | IBM PARAM | | | | | |

The returned IBM PARAM value should be stored in the file control block at offset 16, for subsequent access. The CPM PARAM is assumed to be zero.

**WRITE** - Write a 128 byte record to a file which has previously been opened or created.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | ENQ | | | | | | | |
| | 2 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| | 3 | IBM PARAM | | | | | | | |
| | 4 | LSB RECORD NO | | | | | | | |
| | 5 | MSB RECORD NO | | | | | | | |
| | 6 | CR | | | | | | | |
| ACKNOWLEDGE BLOCK | 1 | ACKH | | | | | | | |
| | 2 | ACKL | | | | | | | |
| | 3 | CR | | | | | | | |
| RETURN DATA | 1 | DATA BYTE 0 | | | | | | | |
| | 2 | | | | | | | | |
| | 3 | | | | | | | | |
| | 128 | DATA BYTE 127 | | | | | | | |
| | 129 | CHECKSUM | | | | | | | |

The user must convert the extent (EX) and current record (CR) fields of the fcb into a four digit hexadecimal number which is transmitted in the request block together with the IBM PARAM value.

When the acknowledge is received the 128 byte record block is sent to the drive unit in the return data block.

If checksum failure is detected the data will not be used to update the file. The checksum value calculation is the ones compliment of the eight bit addition of the data bytes.

Important - Information is only permanently recorded of the file when a CLOSE function is executed after all data has been written.

**READ** - Read 128 byte record from a file which has previously been opened.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | ENQ | | | | | | | |
| | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| | 3 | IBM PARAM | | | | | | | |
| | 4 | LSB RECORD NO | | | | | | | |
| | 5 | MSB RECORD NO | | | | | | | |
| | 6 | CR | | | | | | | |
| ACKNOWLEDGE BLOCK | 1 | ACKH | | | | | | | |
| | 2 | ACKL | | | | | | | |
| | 3 | CR | | | | | | | |
| REQUEST DATA | 1 | CR | | | | | | | |
| RETURN DATA | 1 | DATA BYTE 0 | | | | | | | |
| | 2 | | | | | | | | |
| | 3 | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | 128 | DATA BYTE 127 | | | | | | | |
| | | | | | | | | | |
| | 129 | CHECKSUM | | | | | | | |

The user must convert the extent (EX) and current record (CR) fields of the fcb into a four digit hexadecimal number which is transmitted in the request block together with the IBM PARAM value.

When acknowledges are exchanged the requested record of 128 bytes is returned.

It is the users responsibility to ensure records are not read beyond the end of the file by utilising information in the fcb.

The CHECKSUM value sent is the one's compliment of the eight bit addition of the data bytes.

**CLOSE** - Close a file which has been opened for write functions.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | ENQ | | | | | | | |
| | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 3 | CP | | | | | | | |
| ACKNOWLEDGE BLOCK | 1 | ACKH | | | | | | | |
| | 2 | ACKL | | | | | | | |
| | 3 | CR | | | | | | | |

The close function causes the directory information for a currently open file to be written to the disk.

It is only necessary to perform a close after a WRITE function has been performed.

Not performing a CLOSE function in such circumstances will result in lost data from the disk.

**FORMAT DISK** - Format disk to and create directory space.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | ENQ | | | | | | | |
| | 2 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| | 3 | DISK SIZE PARAM | | | | | | | |
| | 4 | CR | | | | | | | |
| ACKNOWLEDGE BLOCK | 1 | ACKH | | | | | | | |
| | 2 | ACKL | | | | | | | |
| | 3 | CR | | | | | | | |

The disk size param must be set according to the capacity of the disk to be formatted.

Disk param = 0 for 1.4M disk capacity
1 for 720K disk capacity

Disk directory space and file allocation tables are created to make the disk compatible with MSDOS disk format.

The acknowledge block is only returned when the disk format is complete, which can take several minutes.

Important: Only format correct capacity disks. Formatting low capacity disks to high capacity will lead to unreliable data storage.

**FREE DISK SPACE** - Enquiry to determine free disk space.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | | | | ENQ | | | | |
| | 2 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| | 3 | | | | CR | | | | |
| ACKNOWLEDGE BLOCK | 1 | | | | ACKH | | | | |
| | 2 | | | | ACKL | | | | |
| | 3 | | | | CR | | | | |
| REQUEST DATA | 1 | | | | CR | | | | |
| RETURN DATA | 1 | | | | LSB SIZE | | | | |
| | 2 | | | | MSB SIZE | | | | |
| | 3 | | | | CR | | | | |

The information returned from the disk unit is in two bytes representing a four digit hexadecimal value.

This value is equal to the number of 2K byte blocks of disk space which are unallocated.

**STATUS** - Enquire to the disk for status information.

| | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| REQUEST BLOCK | 1 | ENQ | | | | | | | |
| | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| | 3 | CR | | | | | | | |

| | | |
|---|---|---|
| ACKNOWLEDGE BLOCK | 1 | DISK STATUS |
| | 2 | CR |

The information returned in the acknowledge block contains disk status byte. This byte contains information on the disk drive itself as represented below.

DTSK STATUS BYTE

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Charger Bit | Capacity Bit | Read/Write Bit | | Changed Bit | | | Battery Status |

Charger = 1 if charger is in
0 if not

Capacity - 1 for 1.4M disk
0 for 720K disk

Read/Write - 1 for disk read-only
0 for disk read-write

Changed - 1 for possible disk change
0 no change

Battery status - 0 25% capacity level
1 50% capacity level
2 75% capacity level
3 100% capacity level